The University of Jordan

Authorization Form

I Fatima Abdulelah Abualsaud authorize the University of Jordan to supply copies of my Thesis to libraries, establishments, individuals upon request, according to the University of Jordan's regulations.

Signature: Fatima Abualsaud

Date: 10/1/2011

الجامعة الأردنيـــة
كلية الدراسات العليا

التاريـــخ: / /

نموذج رقم (١٨)
اقرار والتزام بالمعايير الأخلاقية والأمانة العلمية
وقوانين الجامعة الأردنية وأنظمتها وتعليماتها
لطلبة الماجستير

أنا الطالب: فاطمة عبدالرحمن عبدالحميد أبو السعود الرقم الجامعي: (8071836)

تخصص: علم الحاسوب الكليـــــة: كلية الملك عبدالله لتكنولوجيا المعلومات

عنوان الرسالة: Dual Recognition of Multi Script Car License plates
التمييز المزدوج للوحات ترخيص السيارات ذات النصوص المتعددة.

اعلن بأنني قد التزمت بقوانين الجامعة الأردنية وأنظمتها وتعليماتها وقراراتها السارية المفعول المتعلقة باعداد رسائل الماجستير عندما قمت شخصيا" باعداد رسالتي وذلك بما ينسجم مع الأمانة العلمية وكافة المعايير الأخلاقية المتعارف عليها في كتابة الرسائل العلمية. كما أنني أعلن بأن رسالتي هذه غير منقولة أو مستلة من رسائل أو كتب أو أبحاث أو أي منشورات علمية تم نشرها أو تخزينها في أي وسيلة اعلامية، وتأسيسا" على ما تقدم فانني أتحمل المسؤولية بأنواعها كافة فيما لو تبين غير ذلك بما فيه حق مجلس العمداء في الجامعة الأردنية بالغاء قرار منحي الدرجة العلمية التي حصلت عليها وسحب شهادة التخرج مني بعد صدورها دون أن يكون لي أي حق في التظلم أو الاعتراض أو الطعن بأي صورة كانت في القرار الصادر عن مجلس العمداء بهذا الصدد.

توقيع الطالب:................ التاريخ: ١٠ / ١ / ٢٠١١

DUAL RECOGNITION OF MULTI SCRIPT CAR LICENSE PLATES

By

Fatima Abu Al-Saud

Supervisor

Dr. Sami Serhan, Prof.

This thesis was submitted in partial fulfillment of the requirements for the Master's Degree of Science in Computer Science

Faculty of Graduate Studies

The University of Jordan

January, 2011

Committee Decision

This thesis (Dual Recognition of Multi Script License Plates) was successfully defended and approved on 9/12/2010.

Examiniation Committee	Signature
Supervisor: Prof. Sami Serhan Compiler Design	------------------------------
Dr. Imad Salah Computer Networks	------------------------------
Dr. Iman Al-Momani Wireless Networks and Security	------------------------------
Dr. Jehad Al-Saadi Distributed Systems	 ------------------------------

Dedication

This thesis is dedicated to my wonderful parents, brothers and sisters. You have been with me in every step of the way, through good times and bad. Thank you for all the unconditional love, guidance, and support that you have always given me, helping me to succeed and instilling in me the confidence that I am capable of doing anything I put my mind to. Thank you for everything.

Acknowledgement

Though only my name appears on the cover of this thesis, a great many people have contributed to its production. I owe my gratitude to all those people who have made this thesis possible and because of whom my graduate experience has been one that I will cherish forever.

My deepest gratitude is to my advisor, Dr. Sami Serhan. I have been amazingly fortunate to have an advisor who gave me the freedom to explore on my own and at the same time the guidance to recover when my steps faltered. His patience and support helped me overcome many crisis situations and finish this thesis.

Many friends have helped me stay sane through these difficult years. Their support and care helped me overcome setbacks and stay focused on my graduate study. I greatly value their friendship and I deeply appreciate their belief in me.

Finally and most importantly, none of this would have been possible without the love and patience of my family. My immediate family, to whom this thesis is dedicated to, has been a constant source of love, concern, support and strength all these years. I would like to express my heart-felt gratitude to my family.

List of Contents

List of Tables

List of Figures

List of Abbreviations

1-D DFT: One Dimension Descrete Fouriur Transform

ALPR: Automatic License Plate Recognition

HSI: Hue, Saturation, Intensity

LPR: License Plate Recognition

OCR: Optical Character Recognition

SVM: Support Vector Machine

VIN: Vehicle Identification Number

DUAL RECOGNITION OF MULTI SCRIPT CAR LICENSE PLATES

By

Fatima Abu Al-Saud

Supervisor

Dr. Sami Serhan, Prof.

Abstract

This thesis exposes a new approach for License Plate Recognition Systems (LPR). Such applications are used in many traffic systems (highway electronic toll collection, red light violation enforcement, border and customs checkpoints, etc.).

The thesis consists of four major phases: extraction of plate, segmentation of characters, recognition of characters and priority checking. As for the extraction phase, edge detection and image eroding algorithms are used. For segmentation phase, fuzzy logic principles are used. While for the recognition phase, neural networks are used. Finally, the priority checking phase that relies on previously calculated and listed priorities is applied.

This priority checking approach is specially programmed for car license plates that have two identical records with different scripts. Its mechanism takes the advantage of the multi-script characteristic and deals with it as if there are two different images of the same plate.

When being unable to recognize a character in one script, it could be recovered and corrected by recognizing the equivalent character of the other script. Subsequent to carrying out many experiments, a smarter step is performed which might change both non matching characters with the most probably right character.

The basis that the approach works for is a correctly extracted and segmented car license plate; this important basis could be affected by so many factors such as weather, dirt, angle, distance, covering objects and non-official license plates.

Following the application of the priority checking method on an LPR system, using MATLAB software, the proposed approach is implemented and the results were positive and recognitions achieved with more accuracy. The more the extracted and

segmented plate was perfectly performed, the more the priority checking was accurate. Upon carrying out many experiments, it was found that this method increases the accuracy of recognizing a car license plate with at least 14% compared to the result of recognizing it with the same implemented system with the priority checking method excluded.

Chapter 1: Introduction

1.1 Motivation

Shortage in such studies and researches in the Arab region was my motive to thoroughly investigate the topic in this thesis. Indeed, some Arab countries use dual script licenses plates, such as Saudi Arabia, Egypt, Yemen, Bahrain and Lebanon. Also, some emirates of the United Arab Emirates used to have dual script license plates.

Even if installing such license plates was not meant to be used in recognition, we could use this feature in our study. If the proposed mechanism is applied to any of the existing license plates recognition systems, it will supposedly increase its accuracy as this is the main motivation we are trying to get as end result. If the proposed mechanism was not applicable, it will not change or cause any problem.

Accordingly, the most proper use for the proposed mechanism is with the systems that has lower accuracy. Analysis of such systems could be carried out in a way that we could add a function that determines which part of the recognized plate is more accurate.

1.2 Problem Description

There always have been car license plates with dual script, as they mostly advantageous in non-English speaking countries. In addition, there exist so many car license plate recognition systems mostly designed for single script license plates recognition. The advantage of using both scripts in the system proposed in this thesis will be taken to enhance the rate of successful recognitions.

This thesis will present and analyze a License Plate Recognition (LPR) system for Saudi license plates that include two kinds of data that refer to the same record. Saudi Vehicle Identification Numbers (VIN)s are repeated in two languages, Arabic alphabets with Hindi digits and Latin alphabets with Arabic digits.

Saudi Arabian Ministry of the Interior changed the old plates that only contained "Arabic alphabets with Hindi digits" with these new plates to make it possible for non-native speakers and writers in Arabic to submit a complainant about a car to the security authorities or to identify their own cars and to make the plates readable if Saudi cars owners' drive in a foreign country.

The proposed idea will get an advantage of this by implementing a system that recognizes both parts and compare them so we get a higher accuracy if both records are matching. Since Saudi license plates are with fixed format and font, the complexity is supposed to be less than unconstrained recognitions. The only problem is that a system like that may get a varied image qualities and illumination due to light and weather conditions that might affect the quality.

Car license recognition is important in several fields of application. These LPR systems are truly required in the world of security. It is used to control international borders, military base surveillance. These systems are also important for traffic control in restricted areas and are highly important to help in the cases of criminal investigation such as when a car is stolen. These systems are also used in traffic law enforcements, electronic toll collection, port and shipping traffic management, parking lot access control, as a marketing tool and to monitor traffic loads. It is generally required in security systems wherever there is a need of identifying vehicles.

1.3 Objectives and Research Questions

The general objective of this thesis is to spot the light on recognizing Saudi car license plates that have dual scripts. In this thesis the following questions are going to be answered and discussed:

Will this method increase the accuracy?

Will this method lead us to get a dependable security system?

What properties and conditions will affect the results?

Is there an improvement that could be done to get a better result?

What if we skipped the recognition of one part?

What if we just chose the most probably accurate part recognized due to a set of previously calculations and stored rates of hits and misses?

1.4 Thesis Organization

The rest of the thesis is organized as follows:

Chapter 2 will discuss car license plates recognition operations step by step then it will list the most common problems that may face such systems.

Chapter 3 will give a brief discussion about LPR systems background and some related works.

Chapter 4 will explain the proposed system ideas and the way they are implemented.

Chapter 5 will describe the simulation environment and the tools used, and finally the results.

Chapter 6 will conclude and summarize the work and give some future work ideas.

Chapter 2: Car Plates Recognition

2.1 Car Plate Recognition Systems

In this chapter, the main phases of an LPR system will be discussed as shown in Figure 1. These systems typically consist of three main modules: plate localization, character segmentation and character recognition.

After getting an image of the back side of a car where the plate should be hanged on, the image goes through the previously mentioned stages. Although some systems could deal with images of different angles or illuminations some only deals with specific angle or brightness. The quality and size of the image also differs between systems.

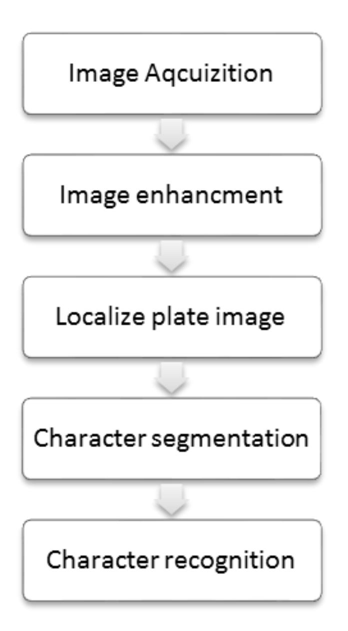


Figure 1 Plate recognition stages

2.2 Image Acquisition

Image acquisition is the first phase in every LPR system. An image of the back side of the car is generally acquired with the following ways:

1. Using a conventional analogue camera and a scanner.
2. Using a digital camera.
3. Using a video camera and a frame grabber (a frame averaging device for optimal frame selection).
4. Digital Camcorder with a Software/Hardware trigger.

The first method is absolutely clear to be impractical for real time image processing applications. In research fields and development purposes a digital camera could be used which can later be exchanged by a camcorder and a frame grabbing device. The last choice requires much programming overhead since the system has to decide between frames taken for the most suitable one to process (Yusuf , 2005). In so many LPR systems today, capturing a license plate is done using specialized camera that is specifically designed for that task.

- **License Plates Features**

Many countries now use license plates that are retro-reflective. That returns the light back to the source and this of course improves the contrast of the image. And in some countries the characters on the plate are not reflective, giving a high level of contrast with the reflective background in any lighting conditions (Constant, 2005).

- **Shutter Speed and Camera Position**

There are two kinds of car plate recognition systems; one is for fast moving cars and this kind need a high shutter speed set to $\frac{1}{1000}^{th}$ of a second so the image captured is not blurry and readable by the Optical Character Recognition (OCR) system. A system that captures images of cars with slow speed does not need a high speed shutter, a shutter speeds set to $\frac{1}{500}^{th}$ of a second can cope with a car moving up to 64 km/h and a shutter set to a $\frac{1}{250}^{th}$ of a second can cope with a moving car up to 8 km/h.

The positioning of the installed camera should be also put on mind. Thus the angle of an image taken can be easily read by the OCR. Exceeding threshold angles of incidence between camera lens and license plate will greatly reduce the probability of obtaining usable images due to distortion. (Ireference, 2010)

2.3 Image Enhancement

An image should be preprocessed before reaching the advanced stages of the recognition system, due to the different illuminations images could have, the contrast and brightness should be changed, the image then gets converted to a black and white one. The image after that is presented as 0's and 1's only.

2.3.1 Histogram Equalization

Usually images have limited contrast; this method enhances an image by equalizing its histogram. A histogram is a plot that shows the frequency of each gray level between 0 (black) and 255 (white). This method generates an equalized histogram which looks

similar to the original histogram although it is stretched to fit across the entire spectrum. (Generation5,2004).This is more clearly shown in Figure 2:

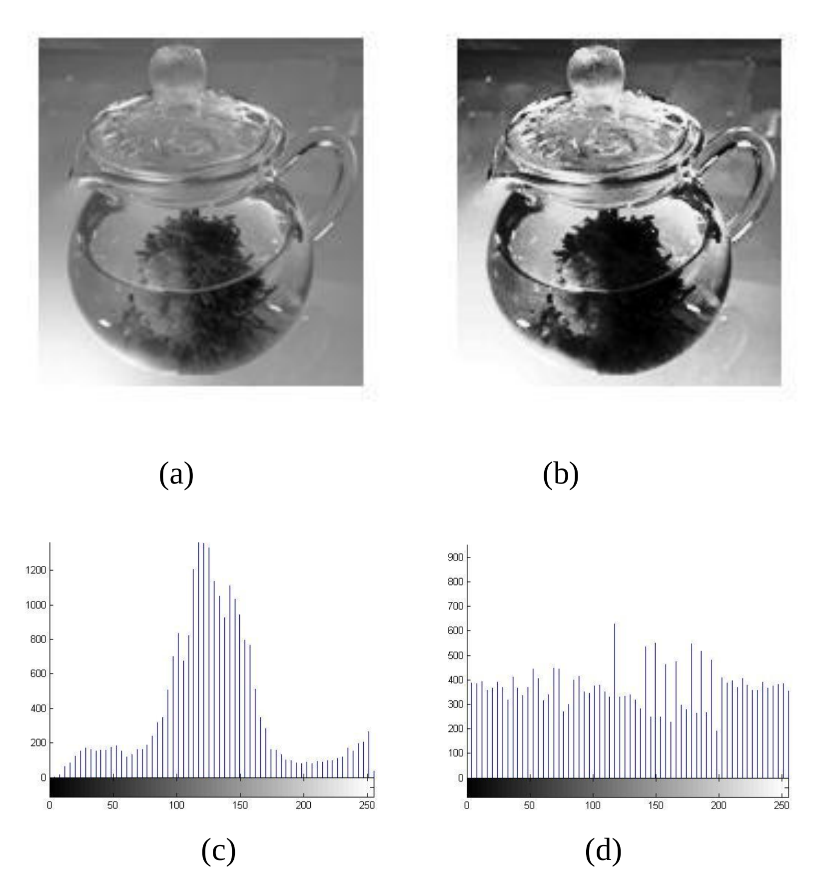


Figure 2 (a) Original image (b) Image after equalizing histogram (c) Original image histogram (d) Histogram of equalized image

2.4 Plate Localization

The license plate may be located anywhere within the image. The plate localization or extraction phase processes the images to detect and extract the region in the image containing the license plate. This can also be considered as an image problem (Parthasarathy , 2006).

2.4.1 Fuzzy Logic Plate Localization

One technique is by using a fuzzy c-means clustering algorithm. And the desired number of clusters is set to two (license plate and non-license plate). The algorithm favors clusters that have a rectangular shape with the correct aspect ratio. (Nijhuis, et al., 1995)

2.4.2 Edge Features

There are two main methods that are used to locate a car plate using edge features, multiple interlacing method and horizontal edge based features method:

- ***Multiple Interlacing Algorithm***

  In multiple interlacing method image is scanned row by row to count edges of each row. Each edge must have a specific distance from another. If number of edges is greater than a threshold, the location of the license plate is guessed in an image. If in the first iteration the program did not find the place, the scan gets repeated with reduced threshold. Repetition is done until place of license is found. (Broumandnia and Fathy, 2005)

- ***Horizontal Edge Based Features***

  In the horizontal edge based features method, the main objective is to locate or isolate an area with highest probability of the presence of the plate. Before doing a plate extraction search over the entire image there must be an improvement to this method by finding some local features describing the presence of a car. This is done to reduce the search space and chances of mistakenly processing a similar structure to the license plate.

  A system had been proposed that searched at first for the following features:

1. Presence of maximum width horizontal edges.
2. Presence of car break lights. (Yusuf , 2005).

Horizontal edges of a car image are always greater than the vertical ones. And this is due to the presence of windscreen joints, bumpers, headlights and other similar objects. Figure 3 shows the horizontal edges produced using Sobel filter.



Figure 3 Horizontal edges using Sobel filter

A 3 x 3 Sobel horizontal edge emphasizing filter h is used, (Gonzalez and Woods, 2002), this filter is shown in the following equation:

$$H = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} * A$$

Mathematically, the operator uses one 3×3 kernel which is convolved with the original image to calculate approximation of the derivative for horizontal changes. If we define A as the source image, H is the image which at each point contains the

horizontal derivative approximation. Where * in the previous equation denotes the 2-dimensional convolution operation.

A local edge-based feature search algorithm is applied to find the widest three lines. Then the search for the candidate license plate is done at first underneath those lines.

A local structure-based search algorithm is applied using the feature of car break lights color, which is red and has the following characteristics:

1. Its invariance to hue in the Hue, Saturation and Intensity (HIS) domain.
2. Dissimilarity of brake lights to any other object in the image (Yusuf , 2005).

2.5 Character Segmentation

Incorrectly segmented characters are not likely to be correctly recognized. In fact, most of the recognition errors are not due to missing recognition power, but to segmentation errors. The limited resolution of plate characters together with dirt, scratches, shadows and skew etc., usually degrades performance of segmentation. In addition, outdoor environment under great variations of illumination changes could make the development of a reliable segmentation scheme very complicated (Tyan & Neubauer, 2002) .

Fuzzy Logic Character Segmentation is used to isolate the individual characters from a license plate using fuzzy logic. At first, applying a global thresholding should be done based on the average gray scale value of the 100 pixels with the largest gradient value. On

the resulting binary image a connected component search is performed. Based on rules concerning the minimal area, width, height of characters, a connected component is marked as a potential character. Any bars between characters or added information that do not obey these rules will be removed.

If the selected components make a valid license plate, then the components are passed to the next stage. A valid license plate differs from country to another, some countries license plates contains six components that starts at the same vertical position. In all other cases the license plate is marked by the systems as unrecognizable and gets rejected. (Nijhuis, et al., 1995)

2.6 Character Recognition

Character recognition is the last step that is taken in almost every LPR system, where this process could be defined as the translation of images of handwritten, typewritten or printed text into machine-encoded text. For example, it is used to convert scanned pages of books or documents into electronic files. The result is a computerized record that makes it easier to search for certain words in a document. In an LPR system such process is done using a technique called OCR.

License plates may be bent and/or tilted with respect to the camera; characters extracted from such license plates may be deformed. Furthermore, input characters may be noisy, broken or incomplete. Character recognition techniques should be able to tolerate these

defects. Optical character recognition methods was developed to deal with the previously mentioned defects (Chang, et al. , 2004) . This approach consists of three steps:

1. Character categorization.
2. Topological sorting.
3. Self-organizing recognition.

In the first step, the input character is distinguished as numerical or alphabetical. This is easily accomplished by referring to the compositional semantics of license numbers. Then the topological features of the input character are computed and are compared with those of pre-stored character templates, these features are shown in Figure 4. Compatible templates will form a test set, in which the character template that best matches the input character is determined. The template test is performed by a self-organizing character recognition procedure (Chang, et al., 2004) .

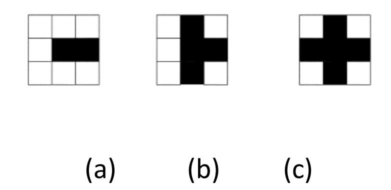


Figure 4 Nodal types: (a) end-point, (b) three-way node, and (c) four-way node.

2.7 Difficulties and Challenges of Such Systems

There are a number of possible difficulties such car plate recognition systems must be able to cope with. These include:

1. Poor image resolution, and this problem might accrue because of two reasons:

    a) Car plate is too far away.

    b) Image was taken with a low-quality camera.

2. Blurry images, particularly motion blur.

3. Poor lighting and low contrast due to:

    a. Overexposure.

    b. Reflection or shadows.

4. An object obscuring part of the plate, quite often a tow bar, or dirt on the plate (Abdalla, 2008); check Figure 5 and Figure 6.



Figure 5 Tow bar that might cover part of the plate



Figure 6 Car plate covered with dirt

5. A different font, some countries do not allow such plates, eliminating the problem.

6. Circumvention techniques, these techniques are used be vehicle owners to evade LPR systems and road-rule enforcement cameras in general to recognize their plates. There are many ways to do so:

   a) Using a plate cover or spray.

   b) Smearing their license plate with dirt or covers to mask the plate.

   c) Cloning or copying registration plates from another car of a similar model and age. This can be difficult to detect.

   d) Foiling infrared detection simply by heating the license plate to a sufficient temperature (Constant, 2005).

7. Lack of coordination between countries or states. Two cars from different countries or states can have the same number but different design of the plate.

A very simple solution is made to solve most of these problems by flagging the unreadable plate so the image gets more attention. And human operators look to see if they are able to identify the alphanumeric of that plate.

Chapter 3: Literature Review

3.1 Background

It is believed that there are currently more than half a billion cars on the roads worldwide. All those vehicles have their VIN as their primary identifier. The vehicle identification number is actually a license number which states a legal license to participate in the public traffic.

To process, sort or analyze data everyone thinks about using computers. If the data is already in the computer most of these tasks are rather easy to be carried out. And because the VIN is the most important information a vehicle can have, a system that needs information about a vehicle to be inserted will absolutely use with its VIN. An LPR system is what we could describe as automation of data input using optical character recognition. Strictly speaking LPR System is an integrated hardware and software device that reads the VIN and outputs the license plate number in ASCII to some data processing system.

Plate recognition systems deal with shots taken for the back side of cars. Images are usually taken from a camera at a toll gate or at a country boarder such as in Saudi Arabian boarders and then preprocessed using several sub-algorithms to reach the final step that recognizes the characters. Almost all plate's recognition systems have a percentage of hits and misses.

3.2 Related Works

Lee, et al. (1994) presented an automatic recognition method of a car license plate using color image processing. The proposed method did not depend on online information of a plate.

Nijhuis, et al. (1995) introduced an LPR system that had been developed to identify VINs for speed-limit enforcement. A combination of neural and fuzzy techniques was used to guarantee a very low error rate at an acceptable recognition rate. First experiments along highways in the Netherlands showed that the system had an error rate, of 0.02% at a recognition rate of 98.51%.

Hegt, et al. (1998) presented an LPR system, which works even under circumstances, which are far from ideal. In a real-life test, the percentage of rejected plates was 13%, whereas 0.4% of the plates were misclassified.

Parisi, et al. (1998) described an experimental system for the recognition of Italian-style car license plates. Images are usually taken from a camera at a toll gate and preprocessed by a fast and robust One Dimension Descrete Fouriur Transform (1-D DFT) scheme to find the plate and character positions. Characters are classified by a multilayer neural network.

Kim, et al. (2000) presents a learning-based approach for the construction of an LPR system. The segmentation extracts the license plate in the detected car image using neural networks. The system recognition module then reads the characters on the detected license plate with a Support Vector Machine (SVM)-based character recognizer. The

system has been tested with 1000 video sequences and has shown a car detection rate 100%, segmentation rate 97.5%, and character recognition rate about 97.2%.

Yu and Kim (2000) propsed a vertical edge matching based algorithm to recognize a Korean license plate from an input gray-scale image. The algorithm is able to recognize license plates in normal and ubnormal shape due to the angle of view. The proposed algorithm was fast.

Sarfraz, et al. (2003) proposed an ALPR system for Saudi Arabian license plates. The system captures the images of the vehicles with a digital camera. The performance of the system has been investigated on real images of about 610 vehicles captured under various illumination conditions. Recognition of about 95% shows that the system is quite effective.

Chang, et al. (2004) proposed an LPR technique that consists of two main modules: a license plate locating module and a license number identification module. The former characterized by fuzzy disciplines attempts to extract license plates from an input image, while the latter conceptualized in terms of neural subjects aims to identify the number present in a license plate. Out of 1088 images, 23 images have been failed to locate the plate; the license plate location rate of success is 97.9%. And out of 1065 images, 47 images have been failed to identify VIN; the identification rate of success is 95.6%. Combining the above two rates, the overall rate of success for our LPR algorithm is 93.7%.

Yusuf (2005) presented an ALPR system for Saudi license plates, in his thesis he proposed techniques based on local car features to reduce the area of search for license plate candidates, image enhancement using intensity adjustment. The algorithms for

extraction phase are based on color edge detection based techniques and candidate plates are extracted using a fuzzy compactness operator and template matching. The segmentation phase stage is performed using Hybrid fuzzy c means (Valarmathie et al., 2009) and image projection profile based technique. The main distinction between the proposed system and his system is that his system uses an old version of Saudi plate's format.

Zheng, et al. (2005) presented a real time and robust method of license plate locating. License plate area contains rich edge and texture information. We first extract out the vertical edges of the car image using image enhancement and Sobel operator (Gonzalez and Woods, 2002), then remove most of the background and noise edges by an effective algorithm, and finally search the plate region by a rectangle window in the residual edge image and segment the plate out from the original car image. Experimental results demonstrate the robustness and efficiency of thier method.

Broumandnia and Fathy (2005) proposeed an automatic LPR system for Persian license plates. They have different type of Persian license plate with different shape, background, font size and structures. The characters are recognized by properly neural network pattern recognition and result stored in database for uses in traffic application. This system worked under variable illumination, variable size of plate and dynamic backgrounds. Using 400 vehicles' images the rate of success recognitions was 95%.

Parthasarathy (2006) presented an ALPR system for United States license plates as an integrated module to assist vehicle emission measurement study. The main distinction is the kind of plates he used which are absolutely different than Saudi new plates. United States license plates have no fixed format, color or font.

Manson (2008) puplished an article that talked about LPR systems that are composed of one or more infrared and color video cameras. They are typically mounted on the top or sides of patrol vehicles in nondescript black boxes. Each frame from the camera's video is processed by a computer that looks for a license plate-sized object in the frame. If a plate is identified, it is analyzed by an OCR software in order to determine its sequence of numbers and letters. This alphanumeric string is then compared against a database of "wanted" numbers. If the plate is not captured or if no "hit" is registered, the system disregards the information and proceeds to look for another plate. If a hit is registered, the officer is notified with an audible and/or visual alert. The entire LPR process is totally automated, requires no officer intervention, and takes less than a second to perform.

Caccia, et al. (2009) presented a paper that describes an approach based on infrared camera and novel methods about how to detect license plates on rear-side of a vehicle in still image or video stream. Particular contribution is posed on discovering plate area by edge search on each side of plate and reconstruction of rectangular shape. The recognized plate area is rotated and adjusted for a better character separation. Top hat morphological operator is used to extract characters from plate background. This operator extracts small elements and details from given images. Each single character inside plate area is separated even in case of tilted shape. This approach try to slice the plate vertically, and it follow the character profile that hit on his vertical path. Pattern matching is used for character recognition.

Carrera, et al. (2009) presented a new method for license plate detection using neural networks in gray scale images. The method proposes a multiple classification strategy based on a multilayer perception. It consists of many classifications of one image

using several shifted window grids. If a pixel belongs or not to the license plate is determined by the most frequent answer given by the different classifications. The result becomes more precise by means of morphological operations and heuristic rules related to shape and size of the license plate zone. The whole method detects the license plates precisely with a low error rate under non-controlled environments.

Through searching for references and as seen above, little amount of study was found made for license plates recognition systems with Arabic script such as in Saudi Arabia. This makes the research and development of such systems broad. LPR systems introduced above were almost all intended for foreign countries and for car license plates with uni-scripts. The following chapter will introduce the proposed work for a LPR system that recognizes a Saudi car license plate with multi-script.

Chapter 4: Proposed Work

In this chapter, details about the proposed system are going to be discussed. The chapter is going to be split into six main parts:

4.1 Image Enhancing

The images are captured using a digital camera from the back side of the vehicles and then inserted as inputs to the system. The primary step that is going to be done to those input images will be enhancing their colors to prepare them for the next steps.

As the new plates are all white colored check Figure 7 (a). The system starts with darkening the colors of the whole image then applying some adjustments to the image such as mapping the intensity of the image by taking certain values of the low-in and high-in as shown in Figure 7 (b). This process will help eliminate unwanted regions through the following stages of the system.

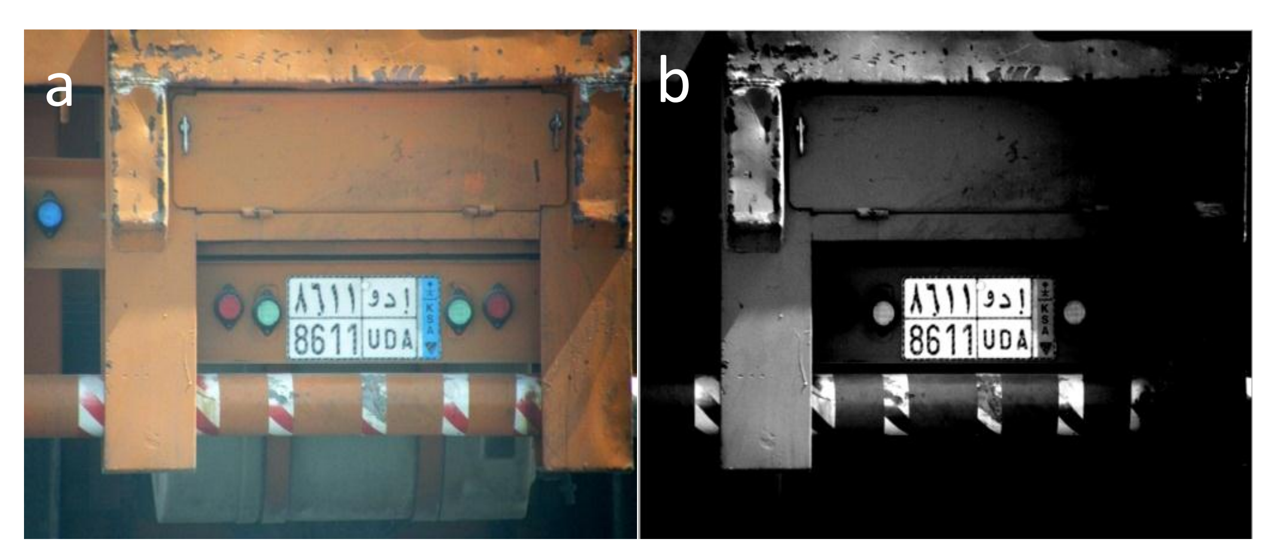


Figure 7 Enhancing input image (a) Original image (b) Enhanced image

4.2 Plate Extracting

Plate extraction step is harder since finding the plate in almost every input image has different conditions. We should take in mind the differences in colors, backgrounds, angles and sizes of the plate. This procedure includes lots of stages that will be discussed in the following steps to get an acceptable result.

1- First step is applying the Binary Gradient Mask (MathWorks, 2010), which gives you the image with all edges. This is found with the Sobel operator and using a threshold of that image, the threshold is used in calculations to define the amount of sensitivity for the Sobel operator check Figure 8(a). The following equation shows the two filters for that operator:

$$G_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ +1 & +2 & +1 \end{bmatrix} * A \quad and \quad G_x = \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix}$$

The resulting gradient approximations can be combined to give the gradient magnitude, using the following equation:

$$G = \sqrt{G_x^2 + G_y^2}$$

2- Finding the Dilated Gradient Mask (MathWorks, 2010), check Figure 8 (b), this procedure will merge thin edges that are next to each other by thickening them. This is done using two flat linear structuring elements that are symmetric with respect to their neighborhood centers, and here are the two elements used:

1 **1** 1

1 0 0
1 0 0
1 0 0

3- Then the image is passed into three sequential functions, one fills the holes check Figure 8(c) and the other clears the borders of the image to omit the unwanted regions check Figure 8(d), third function erodes the image with a given structuring element check Figure 8(e).

4- Then we will create a binary image that contains only the perimeter pixels of objects in the input image, where the perimeter pixel is a nonzero value and connected to at least one zero-value pixel check Figure 8 (f). This will only show the resulted outlines of the previous functions.

5- Eroding function is repeated then with a square structuring element as follows:

1 1 1 1 1
1 1 1 1 1
1 1 **1** 1 1
1 1 1 1 1
1 1 1 1 1

This will result in finding the approximate plate place check Figure 8 (g).

6- Final step will be neglecting the regions that are probably not where the plate is at, this is shown in Figure 8 (h). This procedure is done by eliminating regions containing pixels less than a threshold.

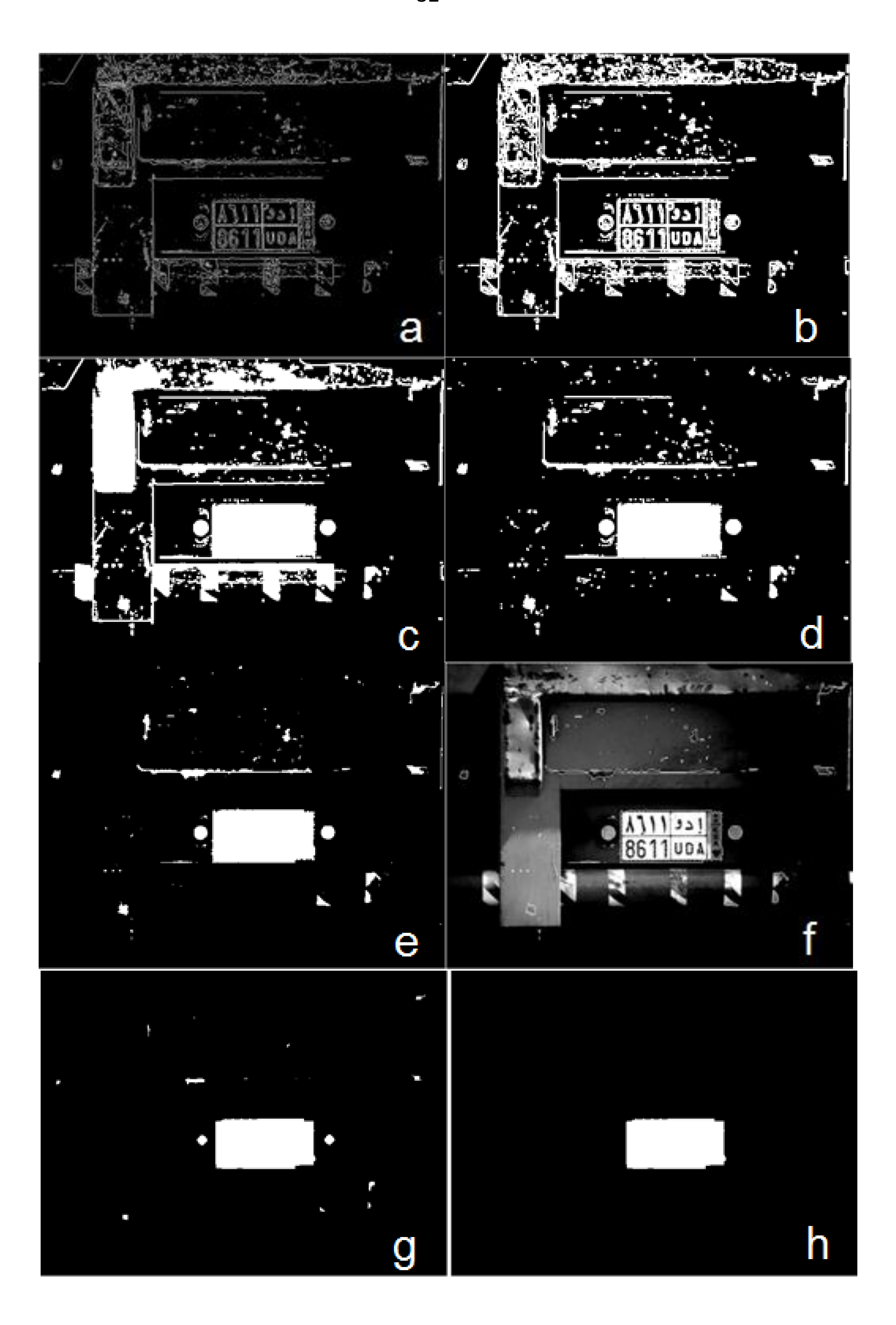


Figure 8 Plate Extraction Process

The following figure shows the resulting extracted plate after applying the previously discussed steps check Figure 9.

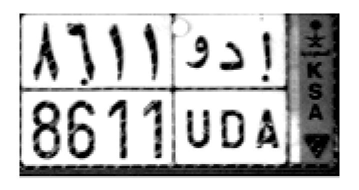


Figure 9 Extracted Plate

4.3 Plate Segmentation

Before passing the extracted plate into the segmentation procedure the system determines in some calculations that are related to particular ratios if the extracted plate shape is wide or thin. And each extracted plate is given a flag to determine that.

The following images show the two shapes of a Saudi license plate, Figure 10 (a) shows the wide shaped plate and Figure 10 (b) shows the thin shaped plate.

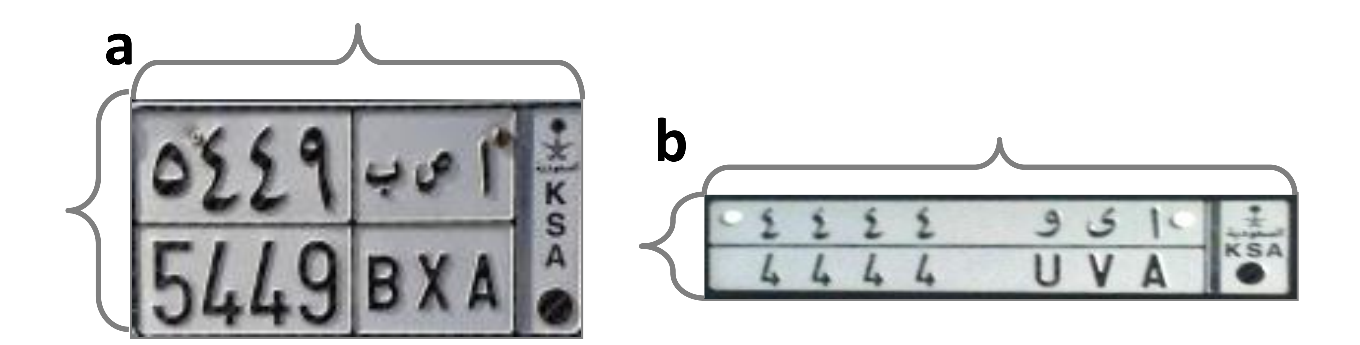


Figure 10 Saudi license shapes (a) wide shape (b) thin shape

The threshold is determined to be 3, where the calculation is done as following:

$$\frac{width(Plate)}{length(Plate)} < 3 \; for \; wide \; shaped \; plates$$

$$\frac{width(Plate)}{length(Plate)} \geq 3 \; for \; thin \; shaped \; plates$$

The following steps are taken to segment each plate:

1- The system begins with checking again for the ratio of an extracted plate which is shown previously by dividing the width by the length. Here, there are two thresholds, one for wide shaped plates and one for thin shaped plates. Those thresholds are used to determine if an extracted plate needs to be trimmed from the right side or not. Right side of each Saudi plate contains a strip with the country logo and the letters K S A, some vehicles such as transportation, diplomatic and temporary cars have this strip with different color (yellow, green, blue, gray and black). Thus, some extracted plates with those colored strip will be already trimmed. To determine that we do the previously described procedure. Figure 11(a) shows an extracted plate that needs the right side to be trimmed and Figure 11(b) shows an extracted plate that does not need to trim its right side.

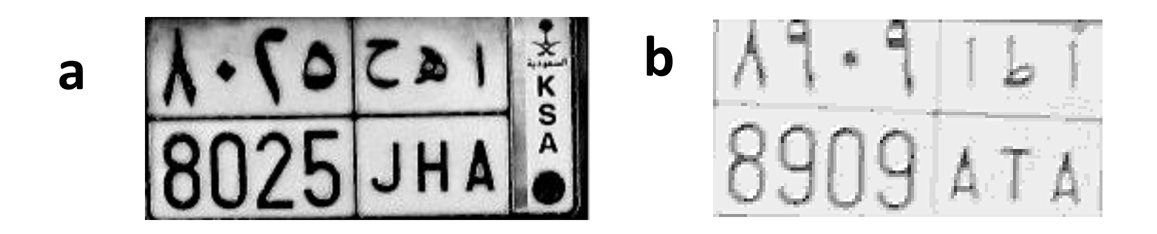


Figure 11 (a) Extracted Whole Plate (b) Extracted Trimmed Plate

A wide plate should be cropped with 1/7 of its width from right side and a thin plate should be cropped with 1/9 of its width from right side. The following equations show the two thresholds talked about in the previous discussion:

$$\frac{Width(Plate)}{Length(Plate)} > 1.69 \ \ wide\ plate\ should\ be\ cropped$$

$$\frac{Width(Plate)}{Length(Plate)} > 4.1 \ \ thin\ plate\ should\ be\ cropped$$

2- All calculations in the segmentation part are done using fuzzy disciplines, each character is supposed to be at the same place in each plate. Thus, the plate will be cut into fourteen segmented part, four Hindi digits followed by three Arabic Alphabets followed by four Arabic digits followed by three Latin Alphabets. This is displayed in Figure 12.

4 Hindi Digits	3 Arabic Letters	KSA
4 Arabic Digits	3 Latin Letters	

Figure 12 Saudi plate characters positioning

3- The segmented parts will go through some processes to get them ready for the recognition part such as resizing and cropping. Figure 13 shows how characters are going to be segmented.

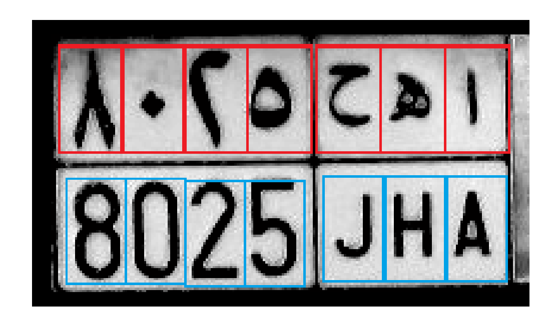


Figure 13 Plate Segmentation

4.4 Character Recognition

This stage is the stage where information is converted from being just an image to a digital record that can be used in sorting or searching. Discussion about character recognition will be divided into two parts:

- **Neural Network Topology**

The system contains a matrix for each alpha-numeric that is 20*20 pixels, which can be represented with 400*1 vectors. The system also contains four neural networks; each neural network has three layers check Figure 14.

Arabic and Latin alphabets neural networks each consists of:

1. An input layer consisting of 400 nodes (for the 20*20 alphabet input).

2. A hidden layer consisting of a number of nodes that was chosen with experiments, for Arabic alphabet neural network there are 35 nodes, and for Latin alphabet neural network there are 49 nodes.

3. An output layer with 17 nodes, each certain node defines certain alphabet.

4. The network uses back-propagation in addition to momentum.

Hindi and Arabic digits neural networks each consists of:

1. An input layer consisting of 400 nodes (for the 20*20 digit input).

2. A hidden layer consisting of a number of nodes that was chosen with experiments, for Hindi digit neural network there are 7 nodes, and for Arabic digit neural network there are 6 nodes.

3. An output layer with 10 nodes, each certain node defines certain digit.

4. The network uses back-propagation in addition to momentum.

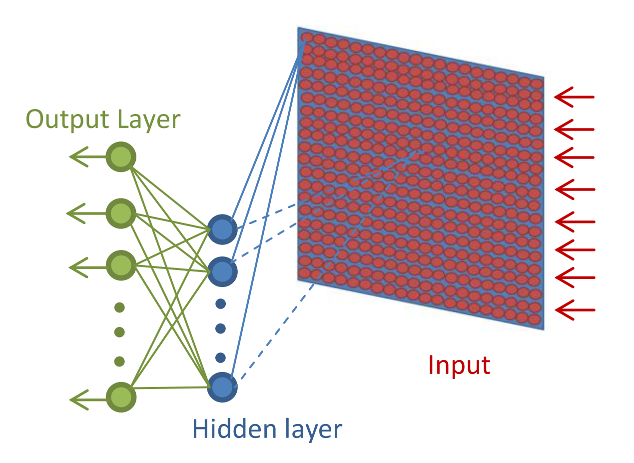


Figure 14Three Layers Neural Network with 400 Inputs

The network is first trained with clear data and the Sum-squared error goal is set to be 0.1 for all neural networks. Momentum constants is used which is equal to 0.95. Then the network is trained again with noisy set of characters.

- **Neural Networks Simulation**

After training all neural networks and making sure all neural networks reached their goals. Simulation is ready to be done by passing each segmented character to its proposed neural network and get the result from the vector that contains all possible candidates.

4.5 Comparing Recognized Parts

This step is the main idea of this work, as we get the advantage of having the two records on the plate. Two vectors, one for Hindi digits and the other for corresponding Arabic digits check Table 1. Two vectors, one for Arabic alphabets and the other for corresponding Latin alphabets check Table 2 . Those two tables are set as references for comparing the recognized parts. By checking for each recognized character with the corresponding recognition if it is matching or not.

Table 1 Matching Digits

٠	١	٢	٣	٤	٥	٦	٧	٨	٩
0	1	2	3	4	5	6	7	8	9

Table 2 Matching Alphabets

ا	ب	ح	د	ر	س	ص	ط	ع	ق	ك	ل	م	ن	هـ	و	ي
A	B	J	D	R	S	X	T	E	G	K	L	Z	N	H	U	V

There are only 17 alphabets used in Saudi plates, plates are designed to contain some of the alphabets only by carefully selecting alphabets that are not similar to each other. Thus, a person who tries to read a plate will not get confused between two alphabets. For example They neglected alike alphabets with only a dot difference.

Figure 15 shows the result of passing segmented plate parts into the neural networks, and as we can see by comparing each character with the corresponding character as in Table 1and Table 2 we can see that the two records are not matching.



Figure 15 Result of Plate Recognition Step 1

4.5 Priority Checking

To deal with the previous problem where comparing the two recognized records did not match, the system has priority for many non-matching conditions such as in Figure 15. For example, the Arabic letter "ر" does not match the Latin letter "D". There are many conditions in this situation:

1. "D" could be changed to the Latin letter "R", this condition has low priority due to the fact that "D" does not look like "R".
2. "ر" could be changed to the Arabic letter "د", this condition has medium priority due to the fact that "ر" might look like "د".
3. "ر" looks similar to "و" and "D" also looks similar to "U". In this situation this condition has the highest priority, thus we are going to change both letters.

By checking for all non-matching characters' conditions we should get what we see in Figure 16.



Figure 16 Recognized Plate Step 2 Priority Checking

- **Priority Checking Pseudocode**

The following steps show the pseudocode for priority checking:

```
Inputs: four Hindi digits, four Arabic digits, three Arabic alphabets and three
Latin alphabets
Outputs: four Hindi digits, four Arabic digits, three Arabic alphabets and three
Latin alphabets

For i=1:4
    IF HindiDigit(i) != ArabicDigit(i)
    function DigitPriorityCheck (HindiDigit(i), ArabicDigit(i))
              IF HindiDigit== HindiValue && ArabicDigit== ArabicValue
                    HindiDigit=NewValue

              END

              IF HindiDigit== HindiValue && ArabicDigit== ArabicValue
                    ArabicDigit=NewValue

               END

    END

    END

 END

For i=1:3
    IF ArabicAlphabet(i) != LatinAlphabet(i)
    function AlphabetPriorityCheck (ArabicAlphabet(i), LatinAlphabet(i))
              IF ArabicAlphabet== ArabicValue && LatinAlphabet==
            LatinValue
                    ArabicAlphabet=NewValue

              END

               IF ArabicAlphabet== ArabicValue && LatinAlphabet==
            LatinValue
                    LatinDigit=NewValue

               END

    END     END     END
```

Arabic alphabets in Table 3 look like each other, in this situation the Latin Alphabet will determine which Arabic Alphabet will choose. In this situation, the corresponding values of Latin Alphabets should not be similar to each other.

Table 3 Similar Arabic Alphabets

Similar Arabic Alphabets	
ق	ب
و	ر
س	ص
ب	ن

Latin alphabets in Table 4 look like each other, in this situation the Arabic Alphabet will determine which Latin Alphabet will choose. In this situation, the corresponding values of Arabic Alphabets should not be similar to each other. For more information check appendices A, B, C and D.

Table 4 Similar Latin Alphabets

Similar Latin Alphabets	
H	N
G	D
K	N
E	B

Chapter 5: Simulation Environment and Results

This section discusses the simulation environment in both hardware and software domains that are involved in the simulation.

5.1 System Hardware

The following points show the specifications of the machine used in the simulation:

1. Intel® Core™2 Duo CPU P8700 @ 2.53GHz 2.53 GHz.
2. Installed memory (Random Access Memory) 4.00 GB.
3. 64-bit Operating System.
4. Digital Camera.

5.2 System Software

The simulation used the following software:

1. Windows 7 professional as an operating system.
2. Matlab 7.0.
3. Microsoft Office Excel 2010.

5.3 Matlab 7.0

MATLAB® is a high-level language and interactive environment that enables you to perform computationally intensive tasks faster than with traditional programming languages such as C, C++, and Fortran.

The system mostly uses the image processing toolbox. The Image Processing Toolbox™ software provides a comprehensive set of reference-standard algorithms and graphical tools for image processing, analysis, visualization, and algorithm development. You can restore noisy or degraded images, enhance images for improved intelligibility, extract features, analyze shapes and textures, and register two images. Most toolbox functions are written in the open MATLAB® language, giving you the ability to inspect the algorithms, modify the source code, and create your own custom functions (MathWorks, 2010).

The system consists of several Matlab m-file functions; it uses several built in functions in addition to newly written functions.

5.4 Simulation Demonstration

All main codes were written from scratch, where the system takes an image as an input and outputs the characters recognized. To show a demonstration of how the system works, we could summarize it with three scenarios, a hit and two misses.

First scenario is where a plate is correctly recognized regardless of how many steps it took the system to recognize it.

1. The system directly recognizes the plate correctly.
2. The system recognizes the plate correctly after comparing the two records and setting the priority of the different characters facing each other.

Second scenario is where a plate is for example missed in the extraction phase and the result is either null or a mistaken recognition, Figure 17 shows a car with a physically missing plate.



Figure 17 Non-existing license plate

Third scenario is where a plate is correctly extracted but the recognition is incorrect due to several cases Figure 18 , such as:

1. Dirty plate or covered plate.

2. Blurred image.

3. Dark screws that holds the plate.

4. Titled plate due to the way image was taken or it is hanged improperly.

5. Non official license plate.

6. Differences in contrasts between regions in the same plate due to shadows.



Figure 18 Cases where a plate is missed recognized

In all scenarios we rely on the following assumptions:

1. Image is smaller than 640*480 pixels.

2. Image is either colored or gray scaled image.

3. Plate size must be larger than 2% of the original image size.

4. Image should not be titled.

5. Plate should not be covered by any object.

5.5 Results

This section will show results of previously done experiments with percentages of hits and misses. The result section will be split into three main parts; the parts will be arranged as follows:

1. First part is the neural network training results where the goal was reached.

2. Second part is the results of recognized noisy characters.

3. Third part is the system overall results, where the entire plate is recognized.

5.5.1 Neural Network Training Results

Figure 19 shows the Arabic alphabets neural network training behavior with random noisy inputs, where goal was 0.1 and the performance reached 0.0990081.

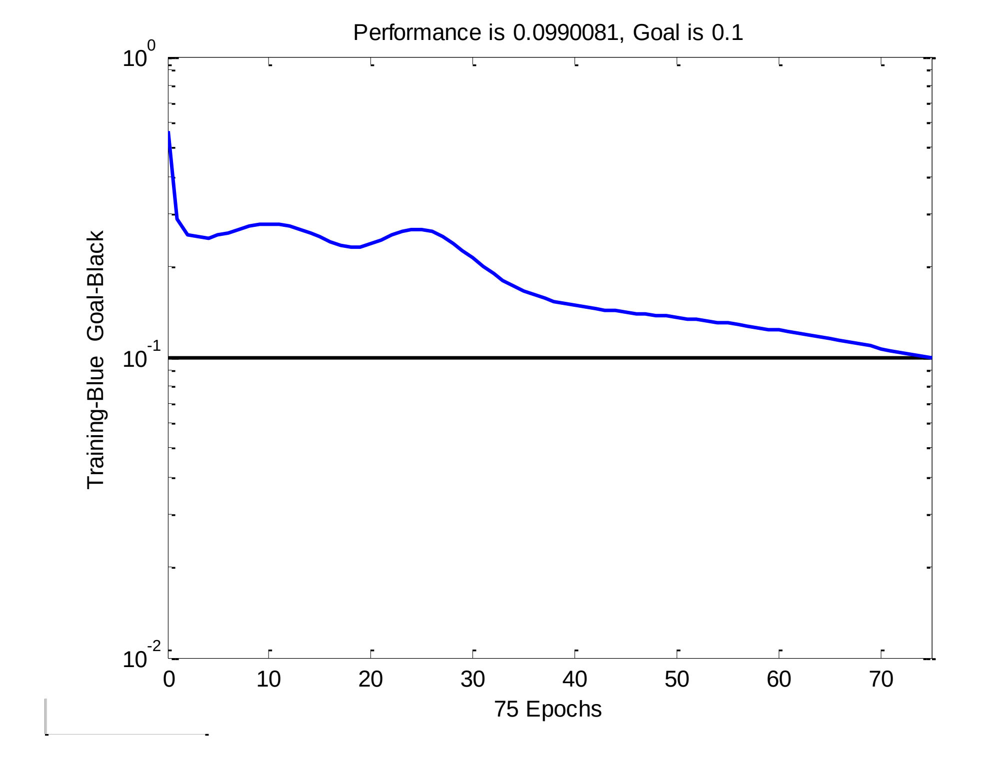


Figure 19 Arabic Alphabets Neural Network Training Result

Figure 20 shows the Arabic digits neural network training behavior with random noisy inputs, where goal was 0.1 and the performance reached 0.0996757.

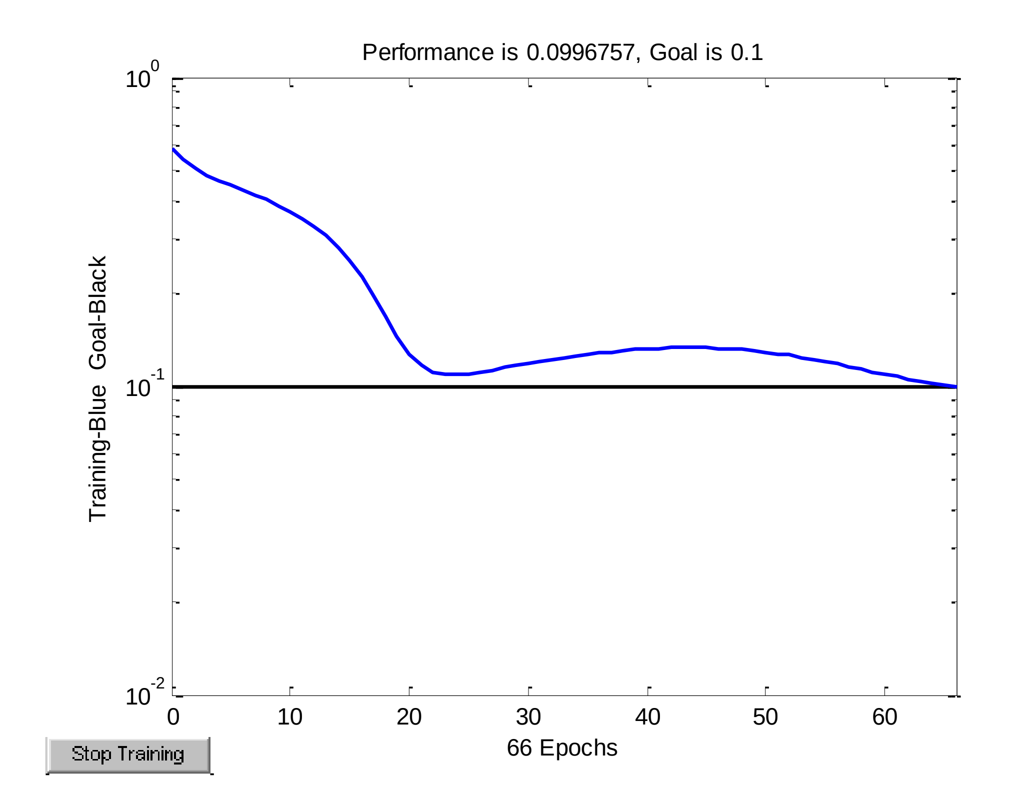


Figure 20 Arabic Digits Neural Network Training Result

Figure 21 shows the Latin alphabets neural network training behavior with random noisy inputs, where goal was 0.1 and the performance reached 0.0991313.

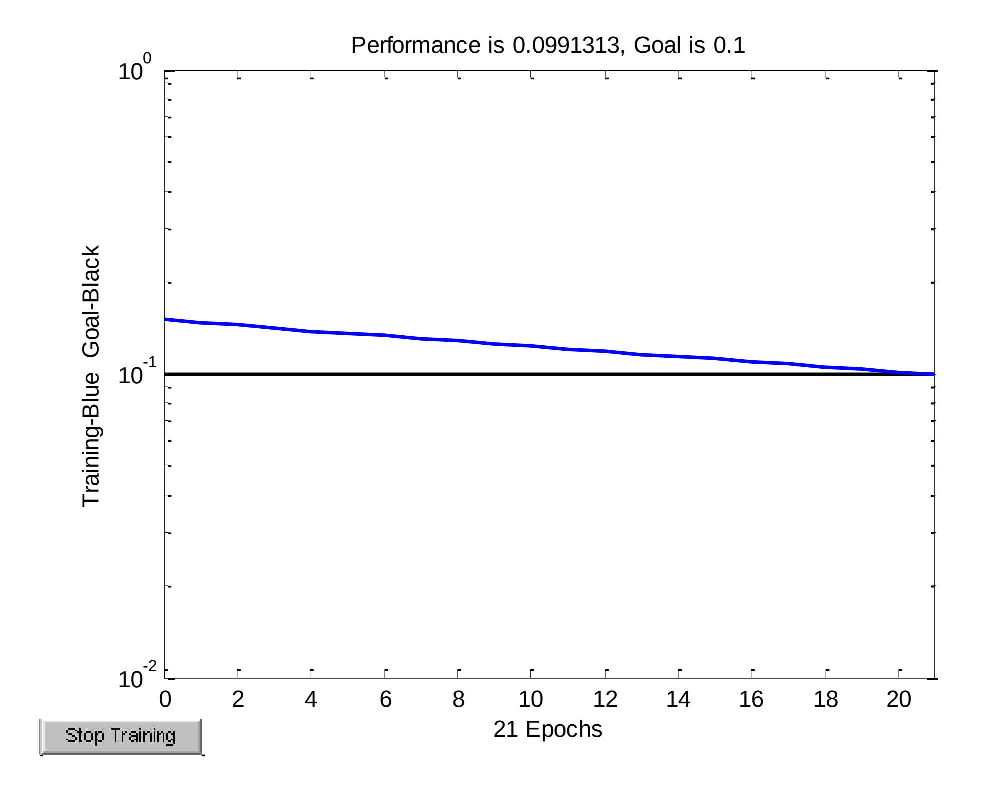


Figure 21 Latin Alphabets Neural Network Training Result

Figure 22 shows the Arabic digits neural network training behavior with random noisy inputs, where goal was 0.1 and the performance reached 0.0999834.

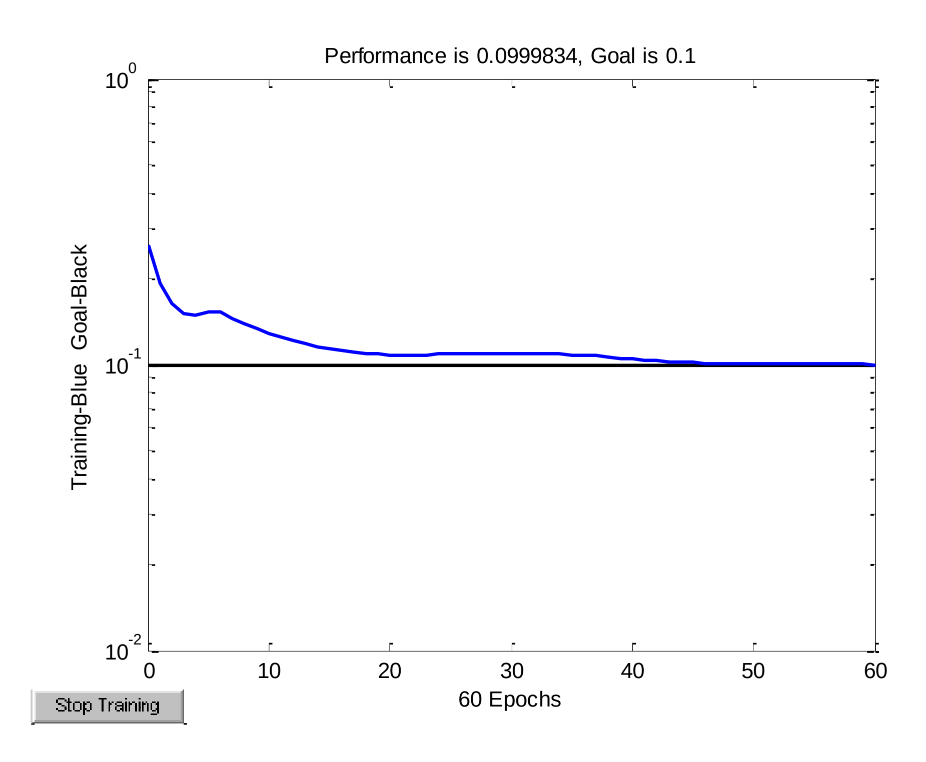


Figure 22 Arabic Digits Neural Network Training Result

Table 5 shows a comparison between all neural network performances and number of epochs till each neural network reached its goal.

Table 5 Neural Networks Performance Comparision

Neural network	Goal	Performance	Epochs
Arabic alphabets	0.1	0.0990081	75
Hindi digits	0.1	0.0996757	66
Latin alphabets	0.1	0.0991313	21
Arabic digits	0.1	0.0999834	60

5.5.2 Results of Recognized Noisy Characters

By adding random values to the original characters and training the networks, then simulating the network with new noisy characters with a lower value of randomness, the following results were found, Figure 23.

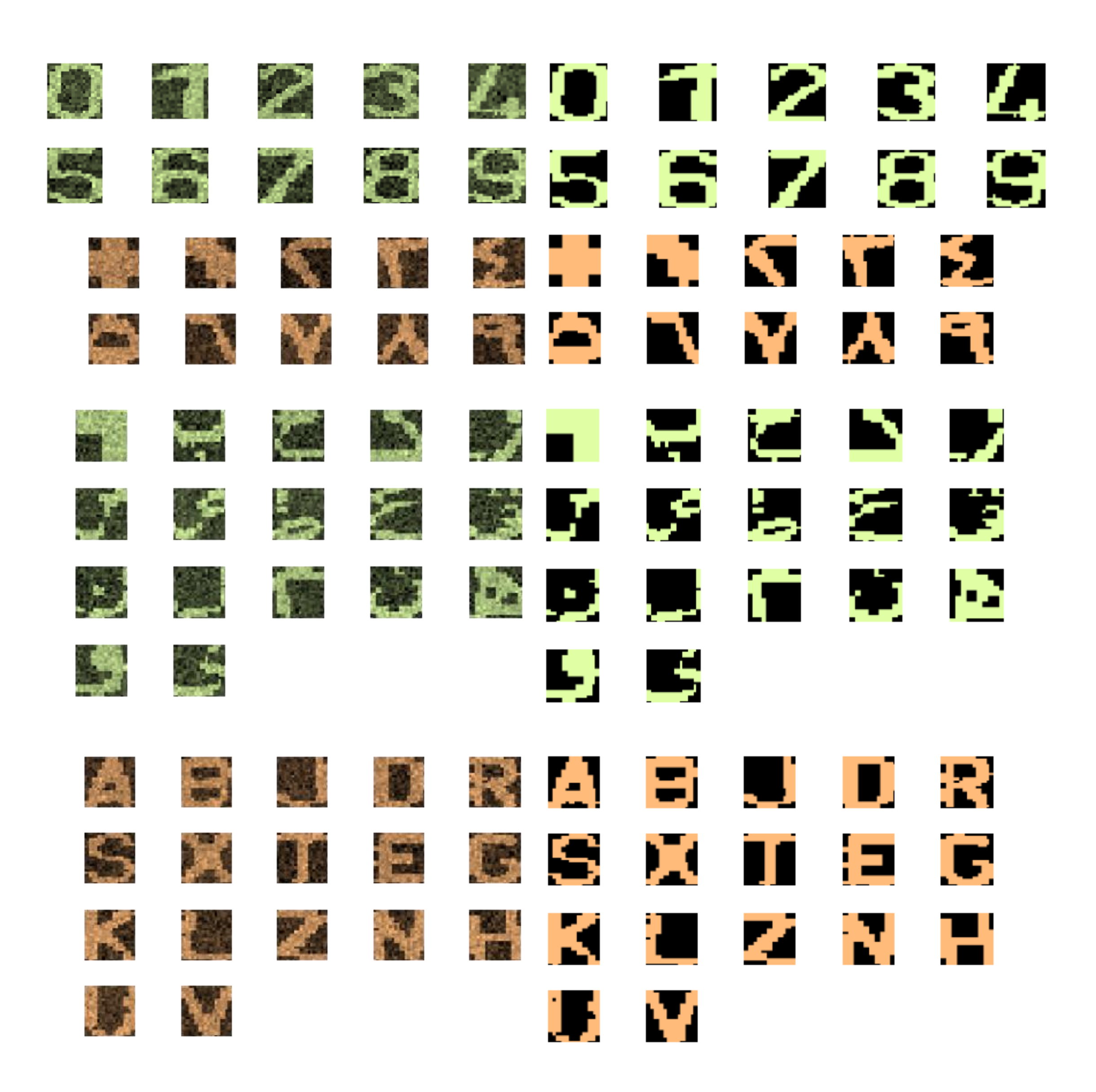


Figure 23 Recognized Noisy Characters

Accuracy was almost 100% in this situation where training was done using random*0.4 and simulation was done using random*0.2. Check Table 6 for more results:

Table 6 Accuracy of Noisy Characters

Noisiness	Arabic Alphabets	Hindi Digits	Latin Alphabets	Arabic Digits
0.2	100%	100%	100%	100%
0.3	100%	100%	100%	100%
0.4	94.1%	100%	100%	100%
0.8	82.4%	100%	100%	100%
1	52%	100%	100%	80%

The previous table shows that the accuracy got affected the most when raising the value of nosiness; this was clearly visible in Arabic alphabets and Arabic digits.

5.5.3 Overall System Results

This section shows the overall results of processing an image through the whole stages of the systems. Recognition is considered a hit if all characters were recognized correctly after passing through all stages.

Tests were done on 75 images and the system proved to be 96% accurate for extraction phase where only 3 plates failed to be extracted, and out of the 72 correctly extracted plates 95.8% were perfectly segmented. Out of the 96% accurate extractions and the 95.8% accurate segmentations, the system reached 69.7% accurate for perfectly recognized plates.

The 30.3% of the missed plates were partially recognized with 85.7% correctness for most missed plates.

Table 7 shows the changes made in the priority phase to seven random plates of the 100% correctly recognized plates; it also shows the percentage of the highest accuracy record in each plate before changing anything:

Table 7 Effect of Priority Checking on Accuracy

Plate number	Accuracy before priority checking	Changes made	Accuracy
Plate 1	71.4%	4	100%
Plate 2	57.1%	5	100%
Plate 3	85.7%	3	100%
Plate 4	85.7%	5	100%
Plate 5	71.4%	4	100%
Plate 6	85.7%	4	100%
Plate 7	71.4%	4	100%

The previously shown results show that the system depends on the records comparison and priority checking phase where the accuracy raised from very low values to 100% accurate record plate. Check Figure 24.

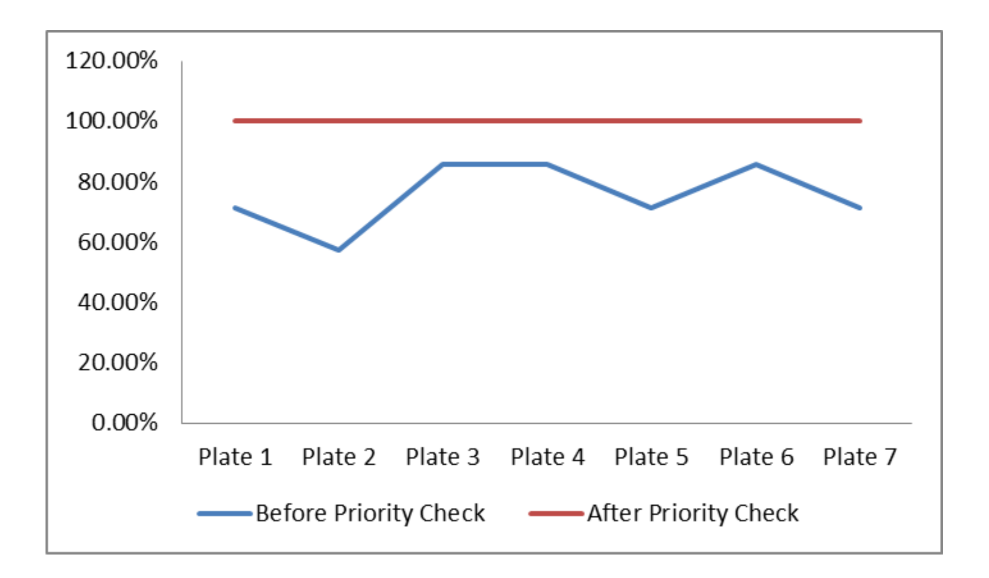


Figure 24 Plot Showing the Effect of Priority Checking

Table 8 shows the changes made in the priority phase to three random plates of the non-correctly recognized plates; it also shows the percentage of the highest accuracy record in each plate before changing anything:

Table 8 Effect of Priority Checking on 3 Non-successful Recognitions

Plate number	Accuracy before priority checking	Changes made	Accuracy
Plate 1	57.1%	3	71.4%
Plate 2	71.4%	4	85.7%
Plate 3	57.1%	2	85.7%

The results above show that the accuracy can really be raised to higher values using the proposed idea. It would give more accurate values in systems with higher accuracy then the system used here. The system showed that it can improve accuracy for plates that are miss-

recognized without the priority check with at least 14% for each missed plate after the priority check, Figure 25.

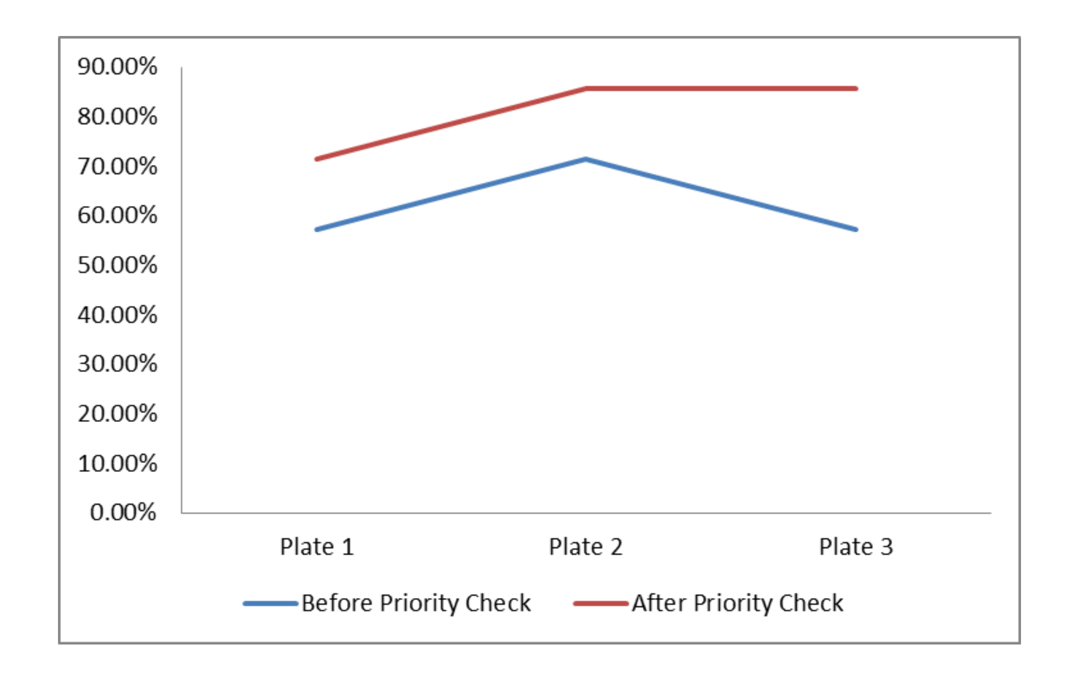


Figure 25 Plot Showing the Effect of Priority Checking on Non-Successfull Recognitions

One of the main purposes of having such an LPR system is for security reasons, this system could be used in traffic systems and those systems have to be reliable. More accurate recognitions lead to more reliable and dependable systems. Such systems then are going to be trusted by lots of organizations and governments.

The priority check method depends on having both parts recognized. With only one part either (Arabic or English) recognized the method is no use. And by checking Table 7 Effect of Priority Checking on Accuracy we would find that results showed that this system recognizes plates using both parts more accurately and by having only one part recognized the accuracy goes down.

Chapter 6: Conclusion and Future Work

6.1 Conclusion

A new approach for car license plate recognition systems has been proposed in this thesis by means of neural networks. The experiments showed that this approach is suitable for low accuracy recognition systems. This approach adds to regular recognition systems a methodology that increases the hits over the misses of a recognized car license plate.

Upgrading the recognition system to use priority checking as shown in chapter 5 showed that:

- The method increased the accuracy by at least 14%.
- The method should lead to more dependable security systems.
- Missing the plate in extracting phase is the main problem that might be encountered and consequently affecting the results.
- Skipping the recognition of one part usually will not lead to better results.

The system was designed for the license plates of the Saudi cars, but could be redesigned for another style of multi-script license plates. The system proved to be 96% accurate for extraction phase, and 95.8% accurate for segmentation phase. Out of the 96% accurate extractions and the 95.8% accurate segmentations, the system reached 69.7% accuracy for perfectly recognized plates. The 30.3% of the missed plates were partially recognized with 85.7% correctness for most missed plates.

6.2 Future Work

As future work, the system should be adapted to deal with the license plates of Saudi cars with less than four digits. Also, more experiments of priorities should be carried out and added to the codes to make the system smarter in finding what characters to change.

The system might be connected to a database for certain groups of cars such as the security systems, to make it easier to find the most probably right candidate for a recognized license plate with different records (a missed plate).

The system might also be connected to another system that recognizes the logo and color of the car and checks them via a database for the most probably right candidate for a missed plate.

References

Abdalla, A. (2008), Automated License Plate Recognition Technology in Law Enforcement, Retrieved 8 4,2010, from Pipstechnology: **http://pipstechnology.com/images/uploads/library/Automated_License_Plate_Recognition_Technology_in_Law_Enforcement.pdf**

Broumandnia ,A. and Fathy ,M. (2005), Application of Pattern Recognition for Farsi License Plate Recognition**. ICGST International Journal on Graphics, Vision and Image Processing** , 5(2), pp. 25-31.

Caccia, F. Marmo, R. and Lombardi L. (2009), License Plate Detection and Character Recognition . **In ICIAP, Image Analysis and Processing** , 5716, Berlin: Springer, pp. 471-480.

Carrera, L. Mora, M. Gonzalez, J. and Aravena, F. (2009), License Plate Detection Using Neural Networks . **In Distributed Computing, Artificial Intelligence, Bioinformatics, Soft Computing, and Ambient Assisted Living** , Berlin: Springer, pp. 1248-1255.

Chang, S.L. Chen, L.S. Chung, Y.C. and Chen, S.W. (2004, March 1), Automatic License Plate Recognition. **IEEE Transactions on Intelligent Transportation Systems**, 5(1), pp. 42-53.

Constant, M. (2005), An Introduction to ANPR, Retrieved 8 4,2010, from CCTV-Information: **http://www.cctv-information.co.uk/i/An_Introduction_to_ANPR**

Gonzalez, R. and Woods, R. (2002), **Digital Image Processing**, (2nd ed.). New Jersey: Prentice Hall.

Hegt, H. Haye, R. and Khan, N. (1998), A High Performance License Plate Recognition System. **IEEE International Conference on Systems, Man, and Cybernetics**, 5 , San Diego, CA, USA: IEEE, pp. 4357-4362.

Ireference. (2010), Automatic Number Plate Recognition**.** Retrieved 10 5, 2010, from Ireference:
http://www.ireference.ca/search/Automatic%20number%20plate%20recognition/

Kim, K.. Kim, K.. Kim, J. and Kim, H. (2000), Learning-Based Approach for License Plate Recognition. **Neural Networks for Signal Processing X Proceedings of the 2000 IEEE Signal Processing Society Workshop** , 2, Sydney, NSW, Australia: IEEE, pp. 614-623.

Lee, E. Kim, P. and Kim, H. (1994), Automatic Recognition of A Car License Plate Using Color Imageprocessing. **IEEE International Conference Image Processing** , 2 ,Austin, TX, USA: IEEE, pp. 301-305.

Manson, T. (2008), License Plate Recognition. **Law and Order Magazine** , pp. 84-92.

Matthews, J. (2004), Histogram Equalization, Retrieved 10 3, 2010, from Generation5: **http://www.generation5.org/content/2004/histogramEqualization.asp**

Nijhuis, J. Ter Brugge, M. Helmholt, K. Pluim, J. Spaanenburg, L. Venema, R. and Westenberg, M. (1995), Car License Plate Recognition With Neural Networks And Fuzzy Logic. **IEEE International Conference on Neural Networks** , 5 , Perth, WA, Australia: IEEE, pp .2232-2236.

Parisi, R. Di Claudio, E.D. Lucarelli, G. and Orlandi, G. (1998), Car Plate Recognition By Neural Networks And Image Processing. **Circuits and Systems Proceedings of the IEEE International Symposium** , 3 , Monterey, CA, pp. 195 – 198.

Parthasarathy, G. (2006), **Automated License Plate Recognition: A Novel Approach using Spectral Analysis and Majority Vote Neural Networks**, Master's Thesis, University of Nevada, Reno,USA.

Sarfraz, M. Ahmed, M. and Ghazi, S. (2003), Saudi Arabian License Plate Recognition System. **International Conference on Geometric Modeling and Graphics**, London, England ,pp. 36-41.

MathWorks, (2010), MATLAB - The Language Of Technical Computing. Retrieved 11 24, 2010, from MathWorks: **http://www.mathworks.com/products/matlab/**

Tyan, J. and Neubauer, C. (2002, 10 2) Character Segmentation Method For Vehicle License Plate Recognition. Retrieved 9 15, 2010, from Freepatentsonline: **http://www.freepatentsonline.com/6473517.html**

P. Valarmathie, P. Srinath, MV. Ravichandran, T. and Dinakaran, K. (2009), Hybrid Fuzzy C-Means Clustering Technique for Gene Expression Data. **International Journal of Research and Reviews in Applied Sciences**, 1(1), pp. 33-37.

Yu, M. and Kim, Y. (2000), An Approach To Korean License Plate Recognition Based on Verticaledge Matching. **IEEE International Conference on Systems, Man, and Cybernetics,** 4, Nashville, TN, USA: IEEE, pp. 2975-2980.

Yusuf, S. (2005), **Identification of Saudi Arabian License Plates**, Master's Thesis, King Fahad University of Petroleum and Minerals, Dharan, Saudi Arabia.

Zheng, D. Zhao, Y. and Wang, J. (2005), An efficient method of license plate location. **Elsevier Science Inc,** 26(15) , pp. 2431 - 2438 .

Appendices

Appendix A: Hindi Digits Priority Checking Table

Table 9 shows how the Hindi digits take the top priority to change Arabic digits.

Table 9 Digits Checking with Hindi Digits Priority

Arabic digit	**Hindi digit**	Arabic digit changed to
5	**3**	3
0	**8**	8
8	**0**	0
9	**0**	0
0	**9**	9
6	**0**	0
1	**7**	7
7	**1**	1
5	**6**	6
6	**5**	5
3	**0**	0
3	**8**	8
8	**3**	3
4	**0**	0
3	**9**	9
9	**3**	3
6	**8**	8
4	**6**	6
7	**2**	2
7	**3**	3
8	**9**	9
..	**..**	..

Appendix B: Arabic Digits Priority Checking Table

Table 10 shows how the Arabic digits take the top priority to change Hindi digits.

Table 10 Digits Checking with Arabic Digits Priority

Hindi digit	**Arabic digit**	Hindi digit changed to
9	**6**	6
6	**9**	9
2	**3**	3
3	**2**	2
5	**0**	0
0	**5**	5
0	**1**	1
1	**0**	0
6	**1**	1
1	**6**	6
2	**4**	4
3	**1**	1
1	**3**	3
2	**8**	8
8	**2**	2
7	**3**	3
9	**7**	7
2	**1**	1
4	**8**	8
3	**0**	0
6	**7**	7
7	**2**	2
..	**..**	..

Appendix C: Arabic Alphabets Priority Checking Table

Table 11 shows how the Arabic alphabets take the top priority to change Latin alphabets.

Table 11 Alphabet Checking with Arabic Alphabets Priority

Latin Alphabet	**Arabic Alphabet**	Latin Alphabet changed to
Z	أ	A
D	ق	G
G	د	D
B	ع	E
E	ب	B
K	ر	R
R	ك	K
T	ل	L
U	ل	L
L	و	U
U	ح	J
J	و	U
N	ي	V
V	ن	N
D	ب	B
B	د	D
S	ق	G
G	س	S
U	د	D
D	و	U
H	ن	N
..	..	..

Appendix D: Latin Alphabets Priority Checking Table

Table 12 shows how the Latin alphabets take the top priority to change Arabic alphabets.

Table 12 Alphabet Checking with Latin Alphabets Priority

Arabic Alphabet	**Latin Alphabet**	Arabic Alphabet changed to
ر	**S**	س
ر	**B**	ب
و	**X**	ص
ب	**S**	س
س	**B**	ب
ب	**G**	ق
ق	**B**	ب
و	**R**	ر
ر	**U**	و
و	**V**	ي
ي	**U**	و
ن	**B**	ب
ب	**N**	ن
ل	**K**	ك
ك	**L**	ل
ق	**U**	و
و	**G**	ق
ك	**N**	ن
ن	**K**	ك
ط	**E**	ع
ع	**T**	ط
..	**..**	..

التمييز المزدوج للوحات ترخيص السيارات ذات النصوص المتعددة

إعداد
فاطمة عبدالاله أبوالسعود

المشرف
الأستاذ الدكتور سامي سرحان

ملخـــــص

تهدف هذه الرسالة إلى عرض منهجية جديدة لأنظمة تمييز لوحات المركبات (License Plate Recognition Systems (LPR) حيث تُستخدم مثل هذه التطبيقات في العديد من أنظمة السّير (مثل، التحصيل الإلكتروني للرسوم على الطرق السريعة، وتطبيق مخالفات قطع الإشارة الحمراء، ونقاط التفتيش الحدودية والجمركية...الخ).

وتتكون الرسالة من أربع مراحل رئيسة هي: استخلاص لوحة المركبة، وتجزئة الحروف، وتمييز الحروف، وتدقيق الأولوية. أما بالنسبة لمرحلة الاستخلاص، فنستخدم خوارزميات الكشف عن الحافة وتآكل الصورة. ونستخدم مبادئ المنطق المبهم في مرحلة تجزئة الحروف. وبالنسبة لمرحلة التمييز، فإننا نستخدم الشبكات العصبية. وأخيراً، يجري تطبيق مرحلة تدقيق الأولوية والتي تعتمد على الأولويات المحسوبة والمدرجة والمذكورة آنفاً.

وقد تمت برمجة طريقة تدقيق الأولوية خصيصاً للوحات المركبات التي لديها سجلين متطابقين مع اختلاف الحروف. وتستفيد آلية عملها من خاصية تعدد الحروف، بحيث نتعامل بهذه الطريقة مع نفس اللوحة على أنه لدينا صورتين مختلفتين لها.

وفي حال عدم القدرة على تمييز أحد الحروف المكتوبة، فيمكننا استعادته وتصحيحه من خلال تمييز الحرف المقابل له من النص الآخر. وبعد إجراء العديد من التجارب، نقوم بخطوة أكثر ذكاءً والتي من خلالها قد نُغيّر كلا الحرفين غير المتطابقين بالحرف الأكثر صحة.

ويُقصد باستخدام أساس عمل هذه المنهجية الحصول على لوحة مركبات مستخلصة ومجزئة بالشكل الصحيح؛ وقد تتأثر هذه الجزئية المهمة بالعديد من العوامل مثل: الطقس والأوساخ والزاوية والمسافة ومواد التغطية ولوحات المركبات المزورة وغير الرسمية.

وبعد تطبيق منهجية تدقيق الأولوية على نظام تمييز لوحات المركبات، والذي يعتمد على برمجيات MATLAB للتطبيقات الهندسية والرياضية، كانت النتائج ايجابية وتمكنًا من الحصول على درجة

أعلى من الدقة في التمييز. فكلما كانت عملية استخلاص اللوحة والتجزئة تجري بإتقان، ازدادت الدقة في إجراء تدقيق الأولوية. وبعد إجراء العديد من الاختبارات، وجدنا أن هذه الطريقة تزيد في دقة تمييز لوحة المركبة بما لا يقل عن 14% مقارنة مع دقة تمييزها قبل إجراء تدقيق الأولوية.